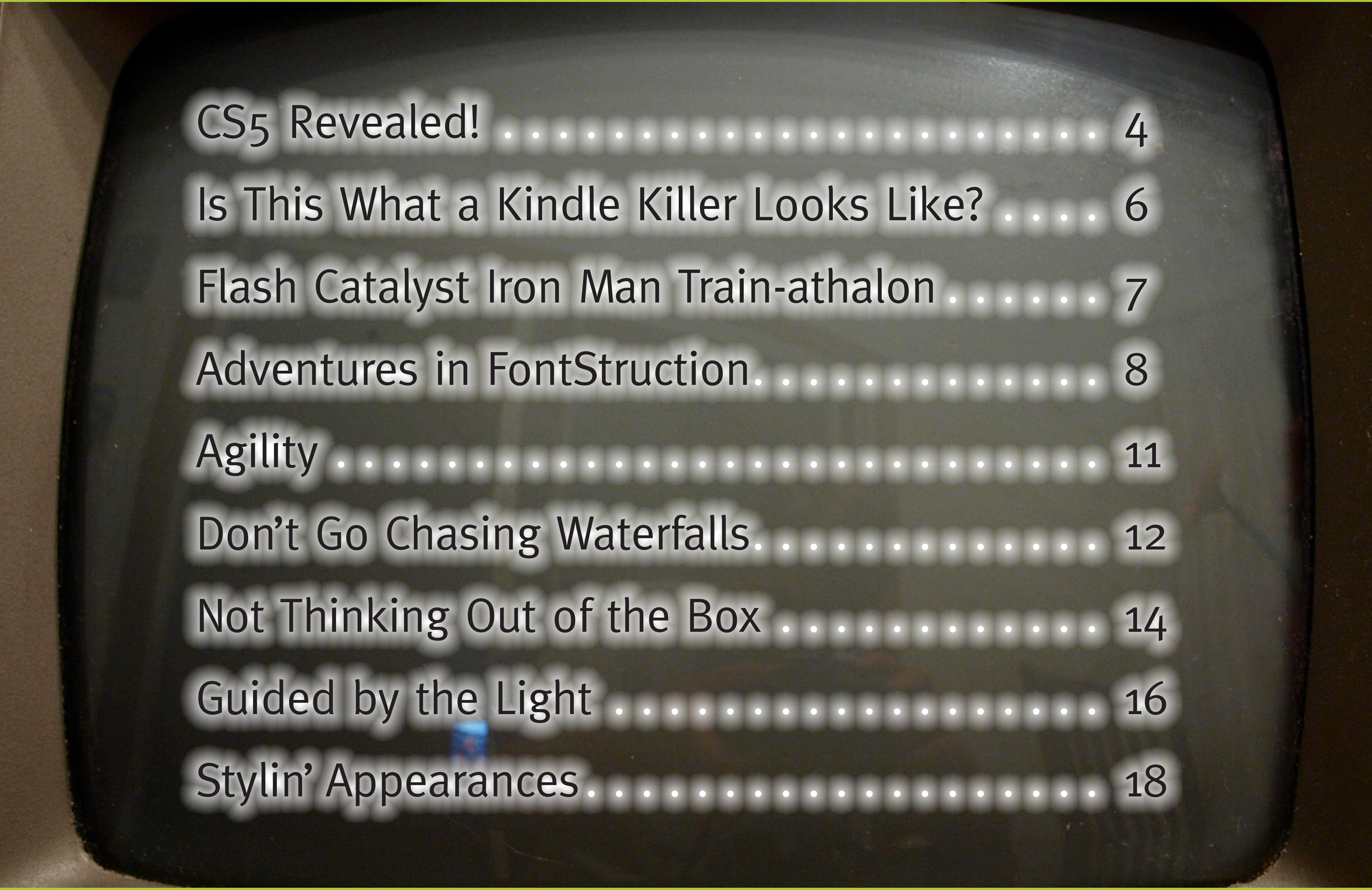
publicious.to.go.07.09

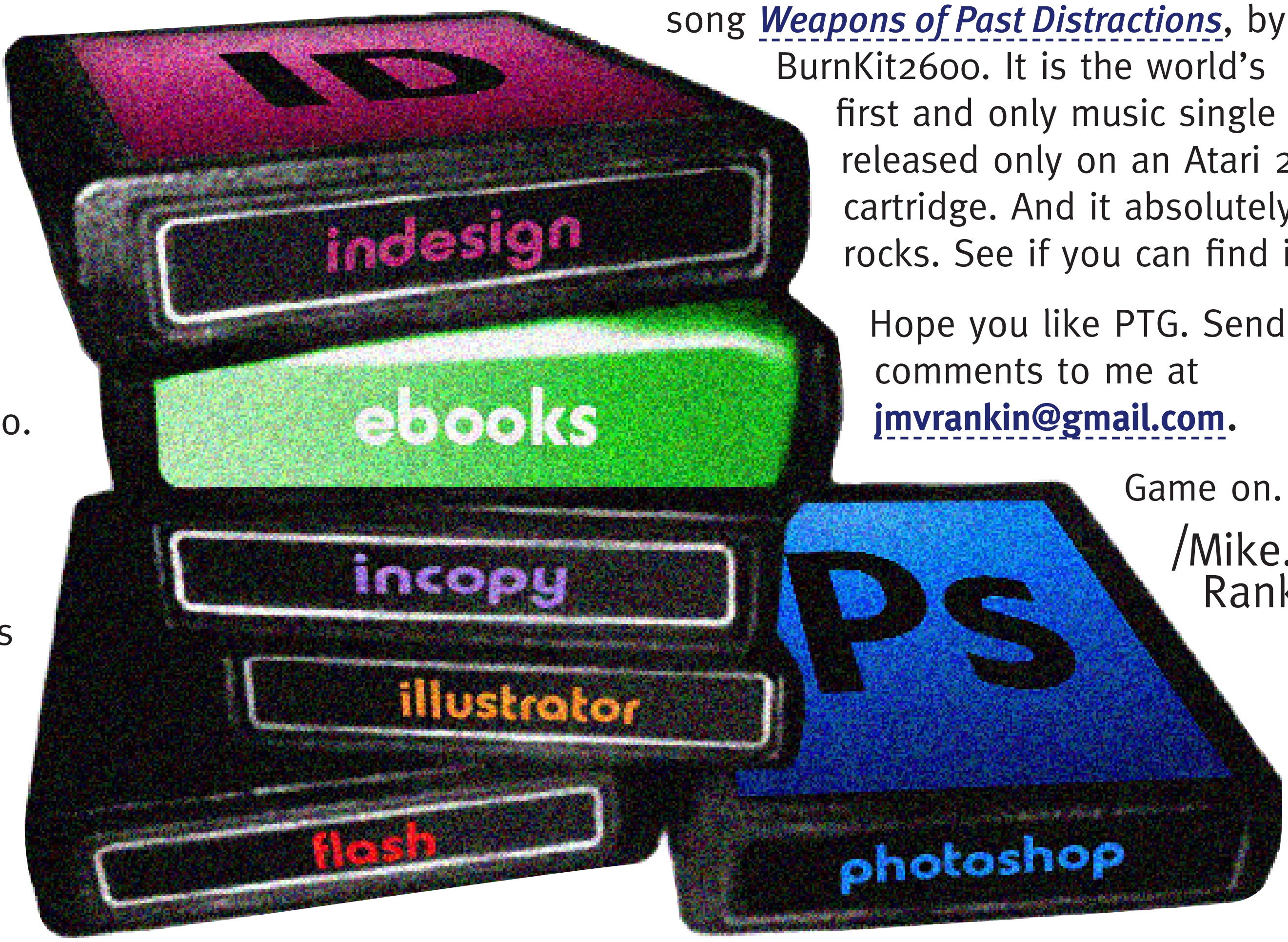


http://

Greetings, salutations, and welcome to the first volume of ***PTG: Publicious To Go.***

Hunting the wilds of the Web for info and entertainment is great, but sometimes you have to gather as well. So henceforth, **Publicious** now comes in a monthly PDF format. I'll try to mix the latest content with some "Greatest Hits" material. And maybe even throw in a surprise bonus or two.

For this first issue, the design theme is vintage video games. I think of what I do for a living as a direct extension of my days wearing out my eyeballs and thumbs with the Atari 2600. The number of bits has grown, but the fun is nearly the same. It's all fun and games until somebody crashes the server.



The "Big Gulp" version of this issue comes with an Easter egg in the form of a hidden soundtrack. Embedded in the PDF is an audio file of the song ***Weapons of Past Distractions***, by BurnKit2600. It is the world's first and only music single released only on an Atari 2600 cartridge. And it absolutely rocks. See if you can find it.

Hope you like PTG. Send comments to me at **jmvrankin@gmail.com**.

Game on.

/Mike.
Rankin

CS5 Revealed!

by Mike Rankin October 5, 2008

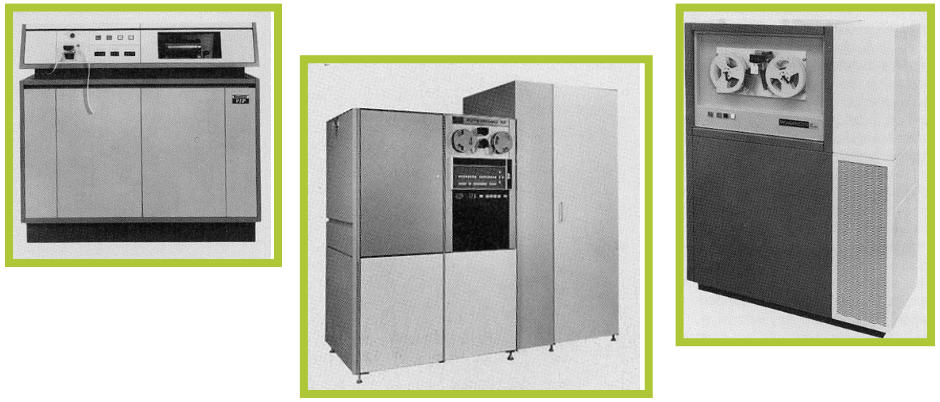
Yes, that's right my friends, CS five! Anybody can show you CS4, but only here at Publicious will you get the scoop on CS5. We have obtained super secret plans from our moles within the Adobe empire, revealing a radical new direction for the Creative Suite.

Codenamed XMullet, the Creative Suite is being reworked from the ground up, with a decidedly retro-shock flavor. In addition, it's clear that Adobe is following in Apple's footsteps, and providing an end-to-end software-and-hardware package. None of CS5 will run on Mac or Windows. Adobe software will run only on Adobe machines. What a gamble! These exclusive photos show just how far out of the box Adobe is thinking. Will the publishing world be ready?

Heavy metal typesetting: adjusting leading with actual lead! It's so obvious; why didn't I think of that?

Here's one of their radical new input devices. Throw away your mice, trackballs, and Wacoms. You'll be able to hold type in your hands.

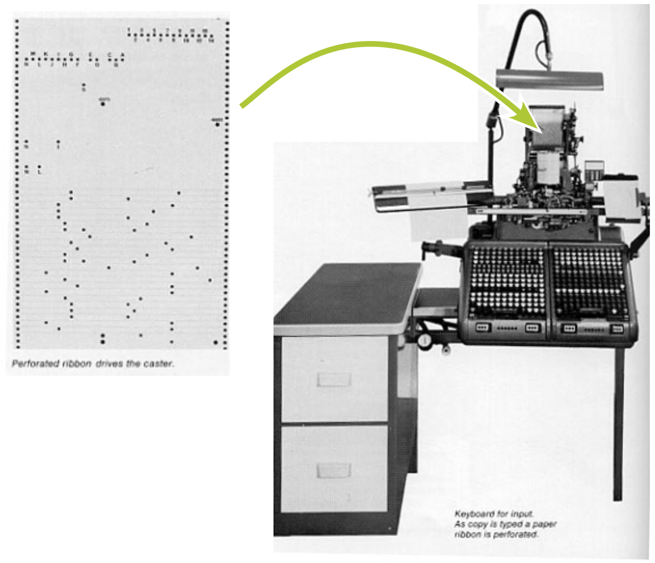
Glimpse at the 3 enormous InDesign Server configurations powering the CS5 revolution.



Actually, I think that thing on the middle right might be a fridge.

Photoshop CS5 will bring a true 3-dimensional workflow. Images are spun around inside a clear drum. Hey buddy, put on some safety goggles!

Behold, Dreamweaver CS5's mind-blowing Code View.



Perforated ribbon drives the caster.

Keyboard for input. As copy is typed a paper ribbon is perforated.

On top of it all, CS5 is totally "green." This Illustrator CS5 workstation runs on diesel, but can easily be converted to run on vegetable oil.

Well, there you have it. What do you think?

I for one, welcome our new Totally Awesome publishing overlords.

Is This What a Kindle Killer Looks Like?

by Mike Rankin   February 11, 2009

One of the coolest things I saw and held at the **O'Reilly Tools of Change conference** was the **Plastic Logic Reader**.

As I played with it, two words came to mind: Kindle Killer. Yes, I know you won't be able to buy a Plastic Logic reader until 2010. Yes, I know Amazon is bigger than the Milky Way, Coca-Cola, and Andre the Giant put together. I also know that what I held was like an iPod and the Kindle, even the Kindle 2, is like a Zune. During the session breaks attendees were swarmed around the Plastic Logic display asking questions and pawing at the thing. I had to trample to two authors and a developer to get my hands on one.

Plastic Logic is positioning the product as more professional and business-oriented than the Kindle, but from what I've seen it's just a more compelling device, period. In my view, the Plastic Logic reader has three killer advantages over the Kindle: size, touchscreen, and file format support. Plus, I'm betting there's an ace in the hole.

The touchscreen technology supports gestures for navigation, annotation, and note taking. You can draw on the screen, and attach notes. The touchscreen also allows for a virtual keyboard. I've never liked the Kindle's look because of the keyboard. Maybe I've been brainwashed from years of iPod UI, but if it's a reading device, the vast majority of the surface area should be devoted to reading.

This also relates to the size issue. The 8.5 x 11 display is much more like what I'd want to have for reading a magazine, news, or a complex work document. I know that makes it less portable, but the Kindle is 8 inches tall, so I'm not sticking that in my pocket either. Plastic Logic has a 150 ppi resolution screen (Kindle 2 is 167 ppi) which can be rotated to display content in either portrait or landscape format. Color capability is planned as well. Here are some **YouTube videos** on the product.

In terms of file format support, Plastic Logic wins too. For reading content, the Kindle 2 supports Kindle (.AZW, .AZW1), Text (.TXT), and Unprotected Mobipocket (.MOBI, .PRC). You can use .PDF, .DOC, and .HTML files only after they have they have been converted to Kindle-readable formats. To convert files you have to either pay Amazon a small fee (ca-ching!), or you have to attach your files to an e-mail that you send to Amazon (privacy? we don't need no stinkin' privacy), and they send you a link to the converted file. Come on. I just want things to work. Period. Plastic Logic supports Office file formats, HTML, EPUB, PDF, and more, out of the box.The claim is that it can display any file you can print.

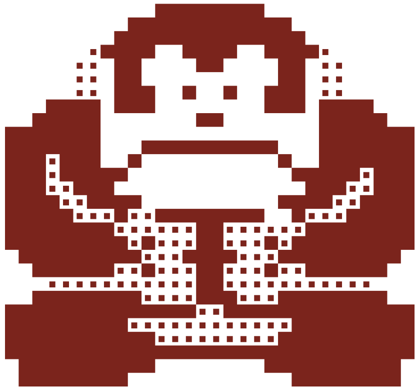
That ace in the hole? Plastic Logic's eReader already has a flexible screen. It's just attached to a hard backing. So it's not too hard to picture a foldable reader evolving from this product. Then you have one killer eReader. Any file format you want, big, color, foldable display in your pocket. Of course, Amazon has walked the walk. The Kindle 2 is out and you can own one. Plastic Logic is still somewhere between drawing board and reality. No word or street date or pricing, but they're off to a very promising start.

PS: Memo to Amazon documentation department, regarding the 100-page **Kindle 2 User Guide**. Thanks for making it readily-available. But if you're not going to put page numbers in the table of contents, for God's sake give me hyperlinks to the pages. Don't make me search or scroll up and down to find where a section might be. Never stop thinking UI, people. Thanks.

Flash Catalyst IronMan Train-athalon

by Mike Rankin June 2, 2009

Back '07, I went to an Adobe MAX conference in Chicago with Eric. Had a lovely time, enjoyed the vocal stylings of Richard Cheese, and was intrigued by a little thing shown in one of the keynotes, called Thermo. It was a tool that would allow designers to work in their designy ways, and build UI for Web applications without having to write code. Filed it away until recently, when the buzz about it became so loud it penetrated the constant meowing of my Siamese cats. Renamed Flash Catalyst, the application is now available for free beta testing on Adobe Labs.



Last weekend I was fortunate enough to attend the "wicked awesome" Flash on Tap conference in Boston (more to come on that!). One of the Adobe keynotes featured a demo of Flash Catalyst (henceforth "FCat" in this post). It looked even cooler than what I had remembered from Thermo days in Chicago. You can use layered AI and PSD files as the basis for your interactive designs. And when the design changes, you can re-open a component in Illustrator or Photoshop and make your changes. That is frigging sweet. But even that wouldn't mean much if FCat generated mucky sucky code. Happily, we are promised that the code FCat generates is rock solid and clean as a whistle. OK, sign me up. Time to start playin'. Just one problem. How do you learn it? Fortunately, with the release of FCat on Labs, there is a flood of tutorials and talk about this new tool.

So, I give you the Flash Catalyst Iron Man Trainathalon. Only the world's strongest eyeballs and mouse fingers will survive.

Start with Kevin Lynch's FCat chat at Web 2.0.

Then go peruse Adobe Labs own list of tutorials.

Click over to Lynda.com for Mordy Golding's freebie introduction to FCat. And for Bonus Mordy, read his essay on MogoMedia.com, Futuristic Hype or Realistic Future?

Back to AdobeTV for FCat videos.

Then hop on your bike and pedal over to the newsstand or library to check out issue 155 of Web Designer magazine, featuring an intro to FCat.

Go to the Help page on Adobe.com, and download the PDF of the Help.

Watch FlashBlog's a 2-part video tutorial FCat.

Then partake of O'Reilly's Inside RIA screencast on FCat.

Chug a RedBull. There are miles to go!

Put on your swimsuit and take a deep dive into Peachpit's lengthy article on FCat.

Then towel off and get ready to climb a mountain: Ryan Stewart's Digital Backcountry blog, for a tutorial in both text-based and video versions. If you're like me, and have some serious Mac OS beachball issues, you might want to check out Ryan's Flash Catalyst performance Tips.

After all that, splash yourself with some cold water and read some posts that are not so enthusiastic about FCat. FlashMagazine's middle-of-the road take on FCat, We Say No To Flash Catalyst Until.., and Catalyst or Catastrophe?

Fortify yourself with a bite from Wikipedia's page on FCat.

Sprint for the finish line: FCat's Facebook and Twitter pages.

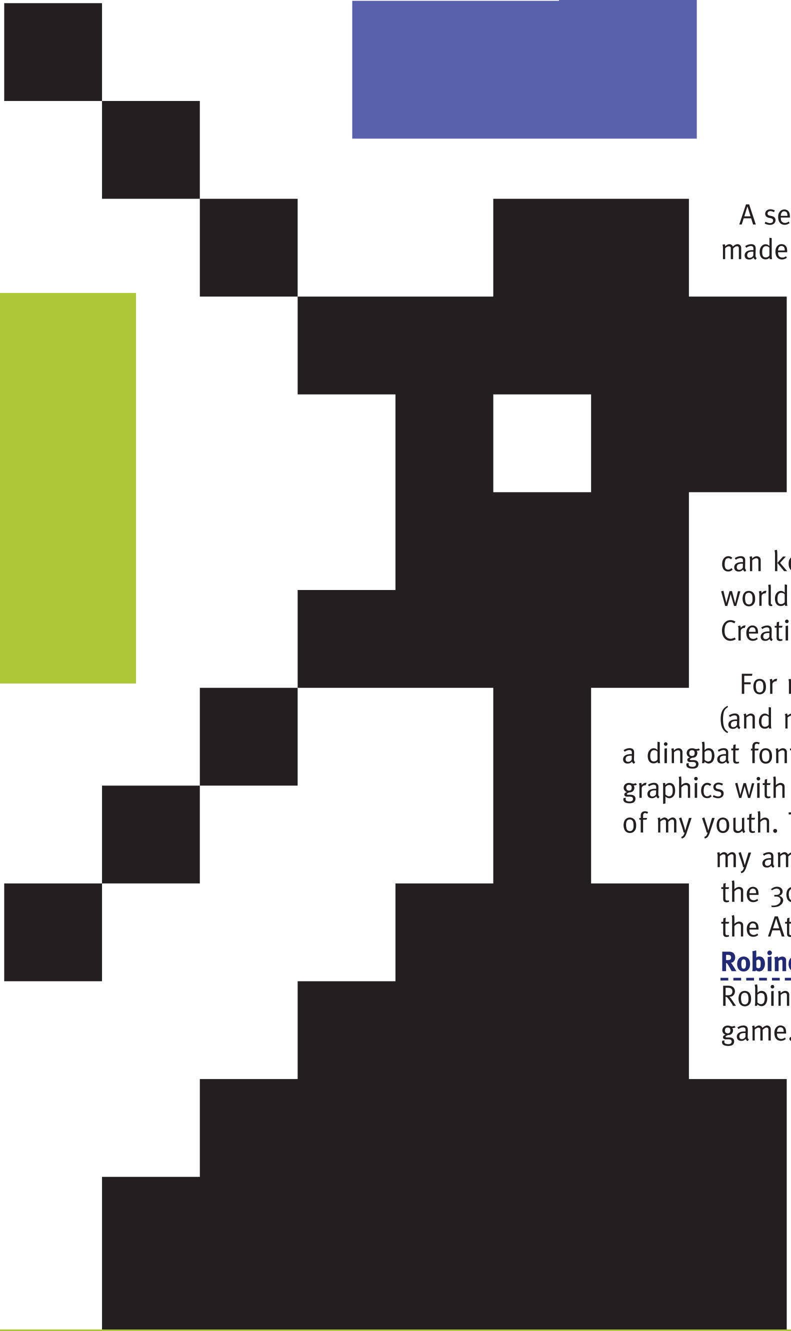
Did you make it? Good! Rest up, kiddo. Tomorrrow we start the ActionScript 3 Steel Cage Learn or Die Death Match.

Adventures in FontStruction

by Mike Rankin May 5, 2008

After a few night's decent sleep, I am now fully recovered from ***l'affaire autosave***. Let us never speak of it again. Time to jump back in the blogfire with both feet. This time the topic is something that's just pure fun. It's my new favorite site/application/addiction, and it's called **FontStruct**.

OK, the name doesn't exactly roll off your tongue. In fact it's impossible to say without sounding like you took a bite of a double-thick fluffernutter sandwich. Don't let that put you off. Call it "HappyMagicFontFun" if you need to. I'll stick with "FontStruct", at least until I sprain a cheek.

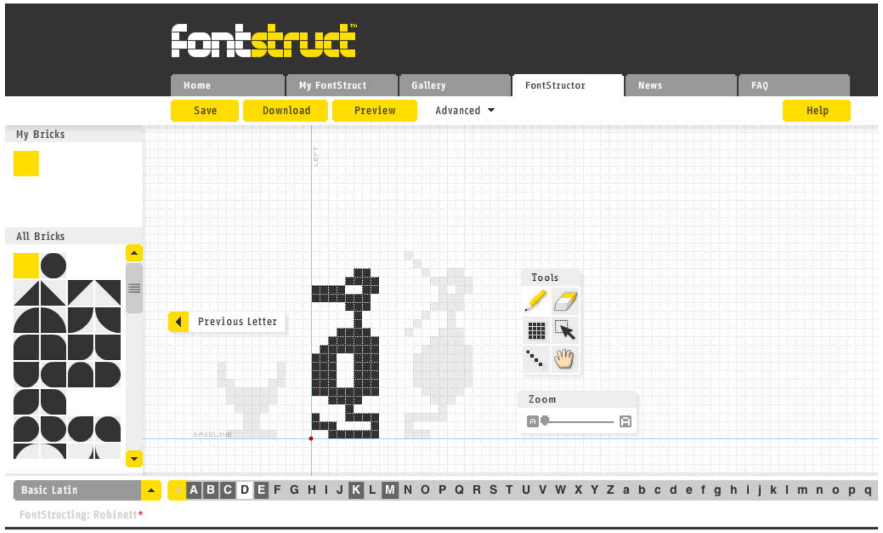


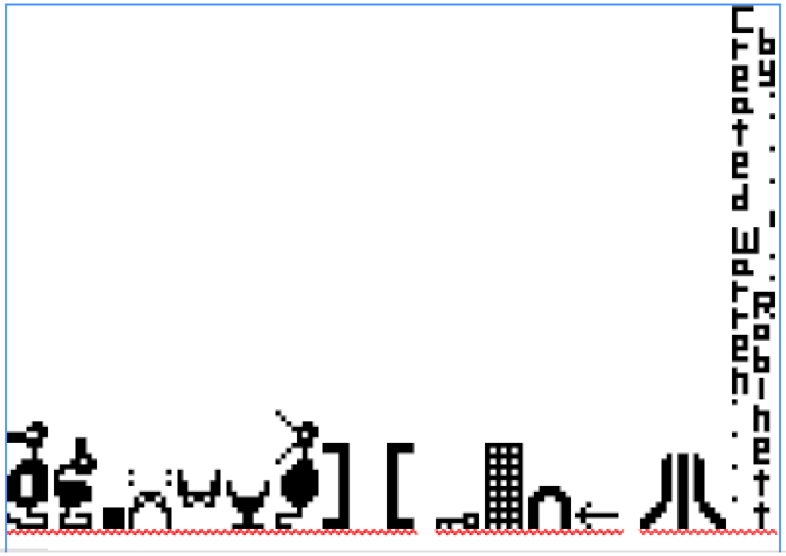
A service of FontShop, FontStruct is a free web app made out of JavaScript and Flash for making your own fonts. It works in the browser, but you can also work offline and save your changes when you re-connect. There's a basic toolset for drawing and editing pixel-based glyphs. You draw with various shaped brushes, called "bricks." It's like Lego for fonts. When you're done you can download your font in OpenType .ttf format (in a zip archive). You can keep it to yourself or choose to share with the world. If you do share, you pick your terms from the Creative Commons license options.

For my first effort, I was inspired by the bricks (and my own need for quick gratification), to make a dingbat font. Something about the idea of constructing graphics with bricks made me think of the 8-bit video games of my youth. Those amazingly simple graphics mesh well with my amazingly simple drawing skills. To commemorate the 30th anniversary of my favorite video game for the Atari 2600, Adventure, I created a font called **Robinett** (after the creator of Adventure, Warren Robinett). It is a collection of the graphics from the game.

I recreated the various dragons, the key, the magnet, the bat, the secret message, etc, bit by bit. I tried to even make a single glyph

fontstruct™





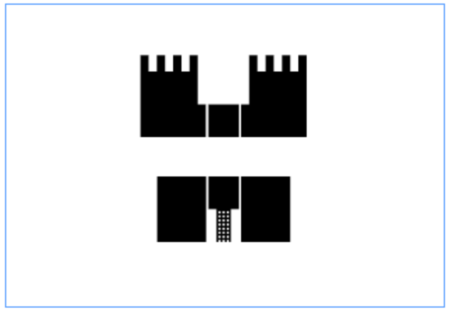
that was a whole castle, but FontStruct wouldn't put up with such craziness. The bigger the castle got, the slower my saves grew, until finally I couldn't save changes at all. In fairness to FontStruct, the full castle glyph contained thousands of pixels.

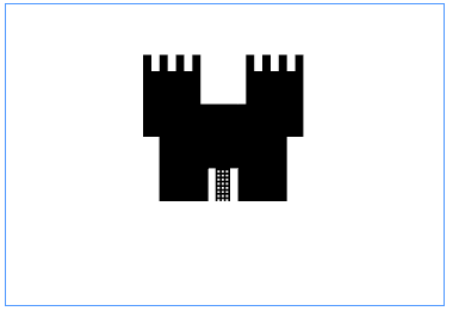
I got around this problem by chopping the castle into six pieces, and attached them to the numbers 1-6. To build a castle you center align, type 123, return, 456. A little baseline shifting and kerning is needed to fit the pieces together snugly.

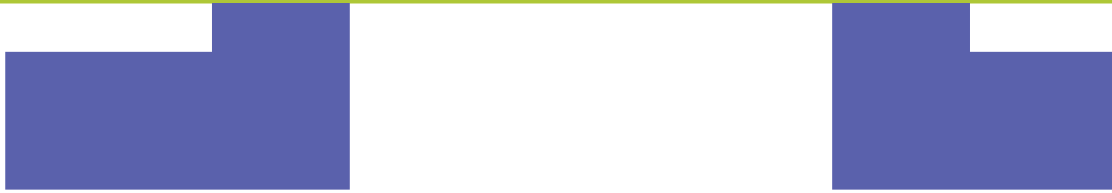
Robinett needs a crazy tall text frame, due to height of the secret message glyph.

Overall, using FontStruct is a very satisfying experience. But I did come up with a few humble feature requests:

- bezier drawing tools
- a way to make non-rectangular selections (i.e. a lasso tool)
- more custom guides
- the ability to upload a JPEG and trace it
- the ability to copy and paste from one font to another
- a beefed up help and FAQ



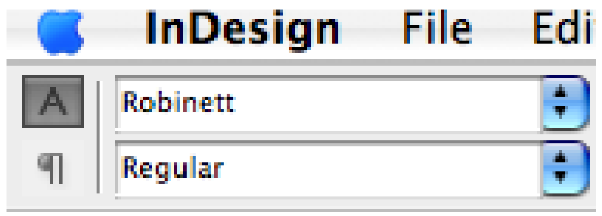


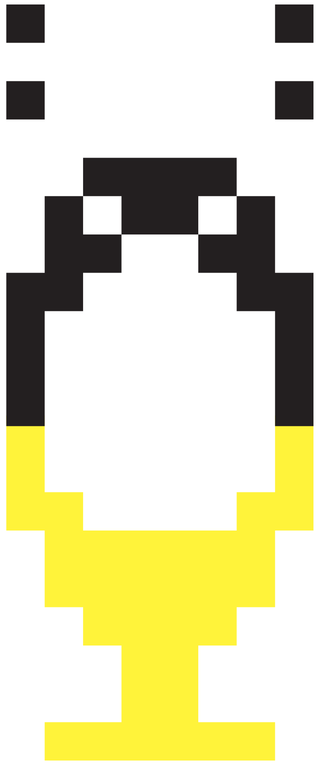


As good a tool as FontStruct is, I think it's even better marketing. FontShop is a company that gets Web 2.0. They hit on every ingredient in the formula for engagement: figure out who your customers are, give them something to free and easy to play with, let them share with each other, be not evil about the legalities, and gently remind users that you also have cool stuff for sale. Your customers become your marketers, driving more people to your site. Hey, it worked on me.

Having this tool at your disposal does not mean you are a type designer any more than having Dreamweaver makes you a Web Designer. There's no "Apply Years of Experience" button or "Keenly Discerning Eye" tool. There's not even hinting to make things nicely kerned. You're free to make a horrible font. And probably a lot of people will.

But constructing, excuse me, FONTstructing an original typeface pixel by pixel is a labor of love. That time committment will weed out a lot of FonTourists. So I think you'll get a lot of true typeophiles, 'structing their stuff for all to see. There's no denying it's cool seeing your font moving in next door to the Goudys and Baskervilles in your typeface menu.





Who knows, if you keep at it, maybe you'll make a future version of the list of the Top 20 Most Important Type Designers of All Time.

PTG Bonus: **Many of the other retro arcade graphics in this PDF are glyphs from another FonStruct font, Old School.**

Not Thinking Out of the Box

by Eric Damitz   April 21, 2009

This is kind of a continuation of the content management stuff I've been writing about. It pertains to software in general though. In our case the problems were with the content management software, but anytime you're developing something based on a commercially-available platform, this issue comes into play.

It's silly, but in our rush to bend our content management system to do what we wanted, we neglected to really explore what it already did. That is, we didn't familiarize ourselves with the "out of the box functionality*" that we were getting.

My main excuse is, of course, the familiar and painfully true: WE DON'T KNOW HOW TO DO THIS. Once again, it would have been nice if our vendor had helped us out a bit more.

One of our team members raised the warning during a UI review-type thing where the vendor showed us an interminable presentation depicting in stunning wireframes the UI representation of each use case. So basically, "if you want to do XYZ, click this button and choose this menu item." Oh, yes, it's just as exciting as you're imagining right now. And did I mention it was in PowerPoint, and we were projecting the slides in a dark room? Probably right after lunch?

Our team member, who wasn't dazzled by the 2-d renderings, asked a simple question along the lines of "what do all those grayed-out menu items do?" You'd think she'd hurled a molotov cocktail at the presenter's head based on the reaction. The presenter, the other reps from the vendor, and our own corporate project manager all basically told her "don't worry about it" but used more and louder words with wavy hand and arm gestures.

Undaunted, she pressed on, asking something like "but don't we need to know everything this does before we can tell you what we don't want?" Again, the arm waving and near-apoplexy with the general message of "we've already figured that out for you based on your requirements."

Oh, my. It pains me to recount this ugly incident. And, like cowards throughout history, I went along with the loud majority, in this case our vendor. Because of course they have our best interests in mind, and they're the experts, and they've assured us that this really is the way we want to go. And the second payment is due next Friday.

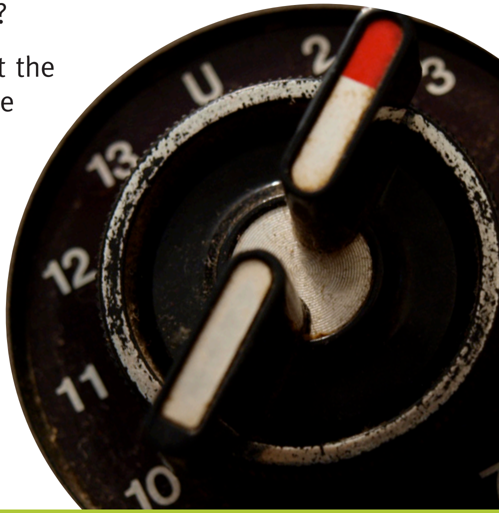
My colleague let it go, but with a "you guys are wrong and this is going to come back to hurt you" look that haunts me to this day.

Obviously, she was right. Months, and even years, later, we'd come across something that didn't quite do what we wanted, or we'd have a new requirement pop up. We'd ask our (completely different) experts how to make it happen. And imagine our anguish when they told us "this is already supposed to be in there." And the absolutely devastating "in fact, somebody had to do a really complex workaround to shut off this core functionality." And the humiliating "why did you want them to do that?"

The lesson should be obvious: if you're using commercially-available software as part of your publishing solution, you'd better have a REALLY good reason to disable features and functions that are already available. I mean, come on. What were we thinking?

(* And may I just say how much I detest the word "functionality**". But even more, the fact that after being bombardized by it for so long I end up engaging in its utilization because it provides optimal expressionality for my core meaningness? This instantiates my angrificity, which renders me beyond uncomfortabilitated.)

(** I did, though, enjoy titling a user guide "Utilizing the Functionality!" The exclamation point means fun!)

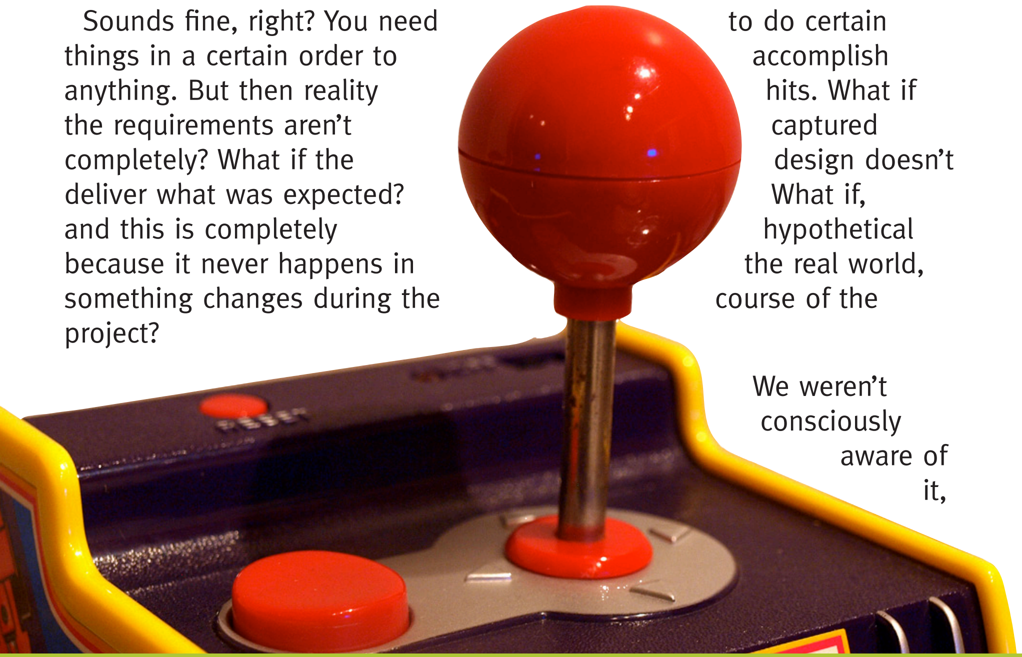


Don't Go Chasing Waterfalls

by Eric Damitz April 28, 2009

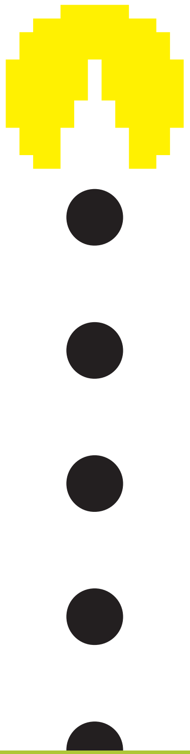
Let's assume that you're going to build an XML-based publishing system. You've gathered requirements from the various stakeholders, have a design in mind, and have vendors or your own people lined up to make it happen. You're good to go. You still have to decide one key thing: how will this system be built?

We really didn't think about this. We came up with a fairly standard method to manage our project. Requirements gathered. Check. Systems designed. Check. Programming implemented. Check. System tested. Check. Boom. Done.



Sounds fine, right? You need to do certain things in a certain order to accomplish anything. But then reality hits. What if the requirements aren't captured completely? What if the design doesn't deliver what was expected? What if, and this is completely hypothetical because it never happens in the real world, something changes during the course of the project?

We weren't consciously aware of it, but we had adopted the waterfall method of project management. The whole project with all its moving parts is considered at once. Everything is developed at the same time. When one stage is finished, that's it, because it's very detrimental to the schedule and budget to revisit past decisions and do rework.



Why is this a problem? Because this programming methodology was developed when guys in white shirts and skinny ties were doing projects on room-size computers that only the richest kings of Europe could afford. You had to have everything planned and ready to go because your computer time was Thursday from 4-5:30. These people did amazing things of course. You try launching rockets to the moon using only punch cards and slide rules.

Of course today, in the impossibly futuristic year of 2009, computers are somewhat more available. There are computers in greeting cards that play stupid songs when you open them. Then you throw them out. My washing machine has more computing power than that room-size thing from 50 years ago.

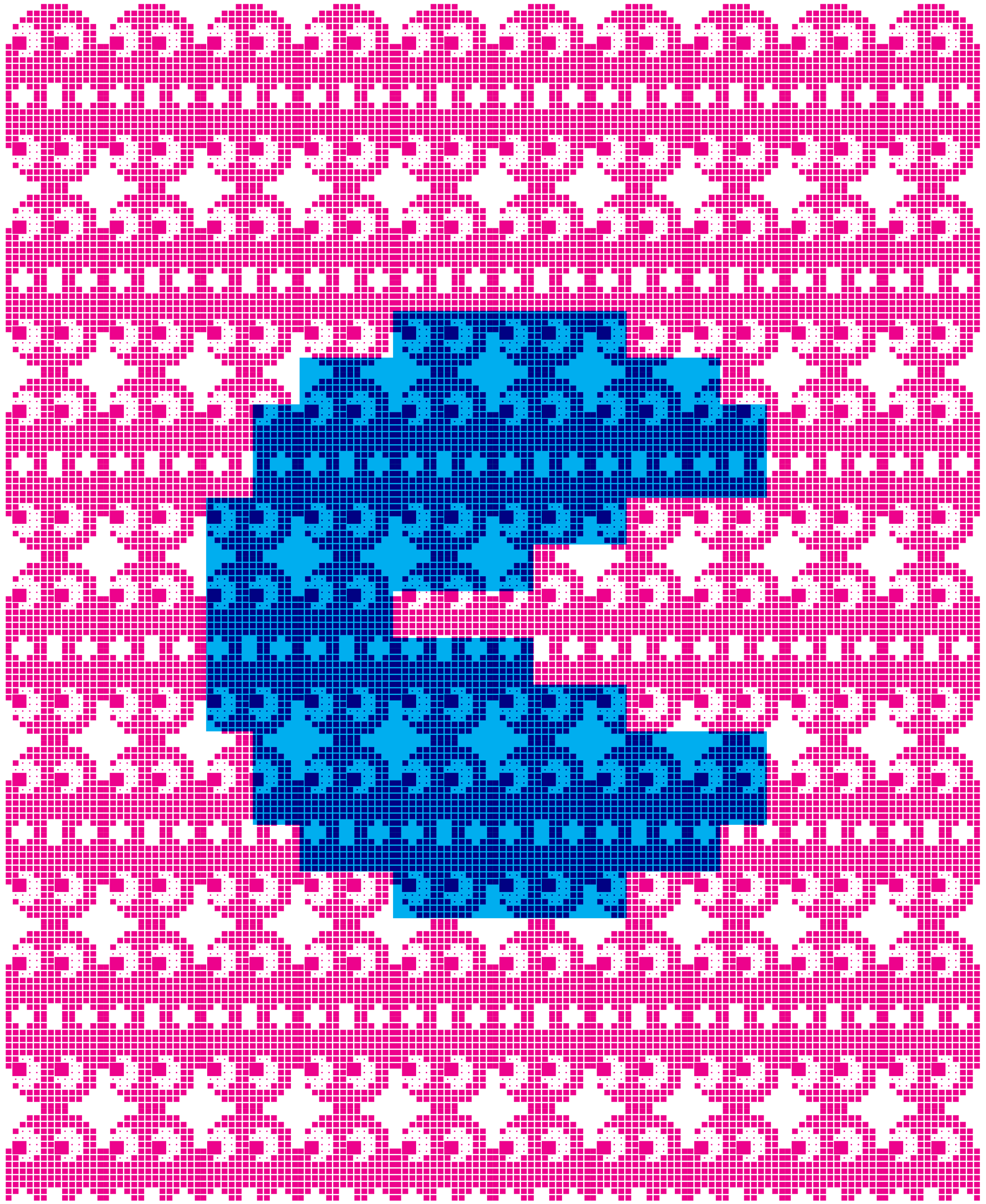
So why would we use such an outdated project management methodology? I don't know. Seriously, who thinks about stuff like this when you're trying to build a hugely complex new thingie on time and within budget? Well, add it to the list of things to talk about while planning a project.

It became clear during development that our methodology was hurting, rather than helping us. The core reality of our existence was, all together now, WE DON'T KNOW WHAT WE'RE DOING. All the requirements and design done before implementation were based on our assumptions and the prior experience of our vendors. Speculation, really. An image of the future that we thought made sense at the time. Something like this.

Then the thing gets built and the testing starts, and we realize we've made horrible mistakes. Gross miscalculations. Silly omissions. And, by the by, the market has changed a bit during all that planning so some of the stuff we planned for and built was no longer needed, and new things that were now needed weren't included because, um, they didn't exist during the planning phase.

Inflexibility is the downfall of the waterfall. It's just too costly to redo the thing once it gets started. Rather than rethinking and reworking major missteps, we were forced to make slight modifications to what we had, which resulted in inadequate solutions and workarounds. Spending all that time and cash, and ending up with something that doesn't quite do everything it needs to do is beyond disappointing.

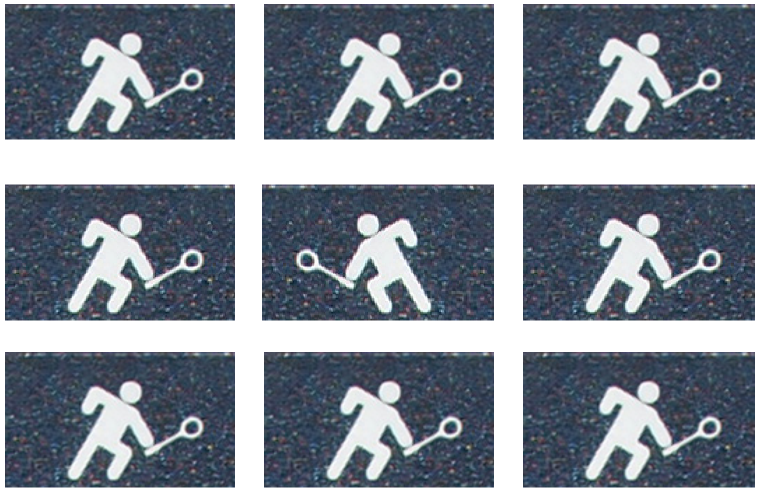
Next week we'll look at another methodology that may help solve some of these problems.



Agility

by Eric Damitz May 5, 2009

The ability to change course as circumstances dictate is the strength of the project management philosophy we'll look at now. **Agile development** is based on the idea that development is an iterative process that uses cross-functional teams to solve problems and develop solutions. Adaptability is the name of the game.



In our current XML publishing project, we've split the initial development phase into several parts. Different teams are focusing on different parts of the system: content modeling, authoring, content management, and composition. Each group is responsible for research and requirements gathering. By splitting the tasks between groups of experts, we're speeding up the whole project, but still are able to delve deeply into each topic.

Contrast this with our previous project, which used the traditional waterfall methodology. In that project, everyone worked on everything. Time and personnel alone imposed limitations on the project. Instead of small groups working in parallel, we had one big group lumbering along, taking on each issue in sequence.

It's kind of like the old christmas-tree lights where if one burned out they all stopped working. What a revelation when they came out with the ones that kept working despite one burned out light.

There are dangers with agile development of course. One of the groups could go off the deep end or down a rabbit hole, or fall victim to any number of cliches as the project wears on. Different groups may have slightly different ideas of what the project is all about. For this reason there still needs to be a strong project manager who's in charge of it all-a benevolent dictator who has the authority to say "no, that's not what we're looking for."

So far, on our project, this methodology is working fairly well. Each group is able to spend ample time researching their area of responsibility. We're getting better information than if we were all lurching around together. It's nice not having to have every aspect of the project in your head at all times. I never learned how to juggle, after all.

At certain points, though, we all have to come together to compare notes and make sure we're on the right track. Inevitably somebody puts together a presentation that gets shown to the management sponsors of the project. Words of encouragement are offered, opinions are aired, and we move on.

As our project continues, I'll be keeping track of any good or bad aspects of agile development. I've never worked on a project to completion using this methodology before. I'm curious to see if it really will help us deliver a better system when we're done.

Guided by the Light
(Blue, Magenta, or even Gray)

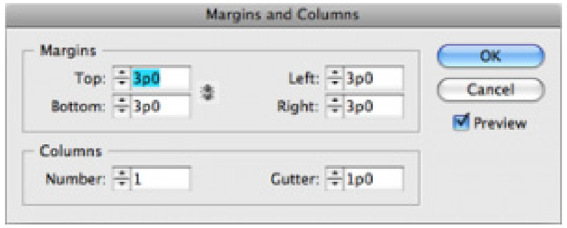
by Cinnamon Cooper April 3, 2009

Guides are a wonderful thing. If you've ever drawn a guide on your layout so you can lock frames to that guide, you'll know how nice it is to just click, drag, let go and get perfectly aligned rows of frames. But sometimes you don't want dozens of guides drawn all over your page.

If you haven't used the Margins and Columns panel under the Layout menu, I suggest you check it out. It's a fantastic way to create guide lines on your layout without having anything that can be accidentally dragged or moved.

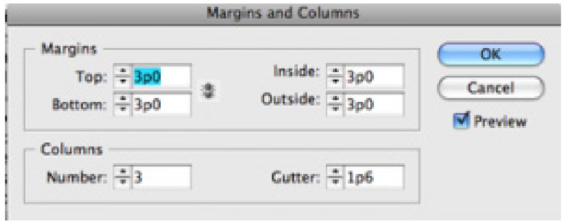
The margin guides can be hidden, are visible in InCopy, and make it a great way to show that you've got your art and text in the live area of your file. Whether you are creating a book where you need to make sure that there is the correct amount of white space in the gutter to keep things from getting cut off in the binding, or whether you're creating a postcard and want to make sure you don't have text too close to the trim so you don't risk losing something that is very important in the final image. And if you're often dragging around frames, if you keep your Snap to Guides selected, your boxes will lock to the margin guides and you'll have less work to perform to get items to align.

If you have a non-facing pages document, your Margins and Columns guide will look like this:



Margins
Top: 3p0 Left: 3p0
Bottom: 3p0 Right: 3p0
Columns
Number: 1 Gutter: 1p0

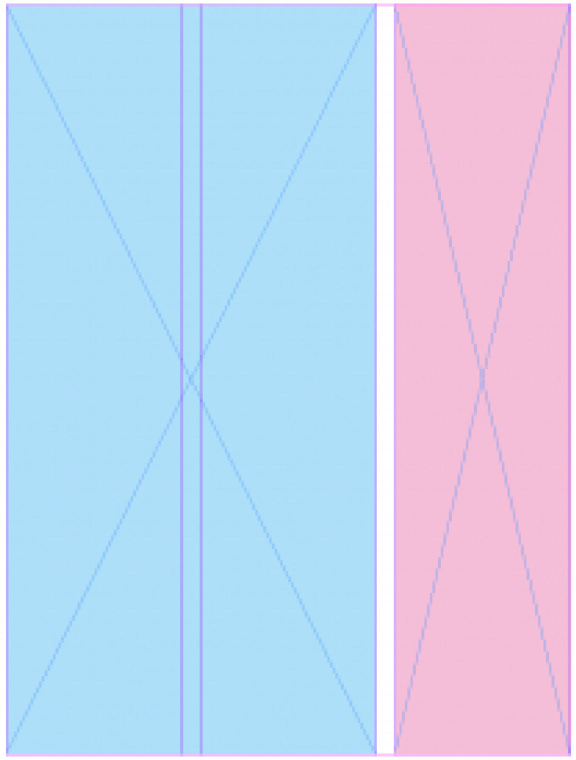
If you have a facing pages document, your Margins and Columns guide will look like this:



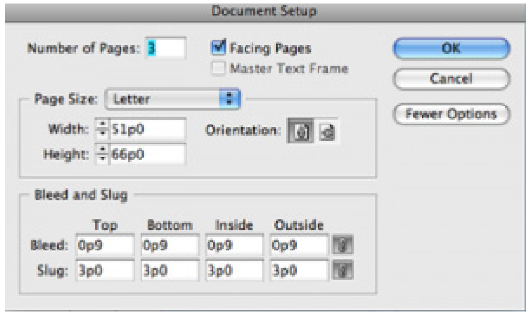
Margins
Top: 3p0 Inside: 3p0
Bottom: 3p0 Outside: 3p0
Columns
Number: 3 Gutter: 1p6

And that's great. Having a guide on the page to always let you know if you're within your safe printing or binding area is wonderful. But there is more that you can do with unselectable guides.

In that same window there are column guides you can set, too. Imagine a newspaper front page. There are likely very even columns of text with white space between them and occasionally image boxes that align on those columns. But what if your layout calls for you to have one really wide column and one skinny column? Well, if you choose 3 columns and adjust the gutter (which is the white space between columns) then you just can create one text area that runs across two columns and one text frame that runs across one. In this example the light blue box is your main text column and the light pink box is your secondary text column.



The Document Setup window also has settings that will apply permanent guides. If you've got documents that need to have a set bleed per your printer's requests, you can set that as well in the Document Setup window. Click on More Options and you can enter in the amount of your bleed and a colored guide line will appear outside your document edge. If you enter a slug area in that same window you'll get another colored line appearing outside your document edge. The slug area is a great place to put information that you may want to be able to include in a printout or a PDF, but that you would also want to exclude from a printout or PDF. (When you print, you can choose to have your bleeds or your slug area included in the printout or PDF.


Document Setup

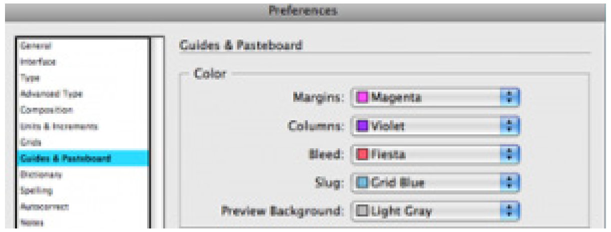
Number of Pages: 3 ✓ Facing Pages, Master Text Frame

Page Size: Letter — Width: 51p0, Height: 66p0, Orientation

Bleed and Slug

	Top	Bottom	Inside	Outside
Bleed:	0p9	0p9	0p9	0p9
Slug:	3p0	3p0	3p0	3p0

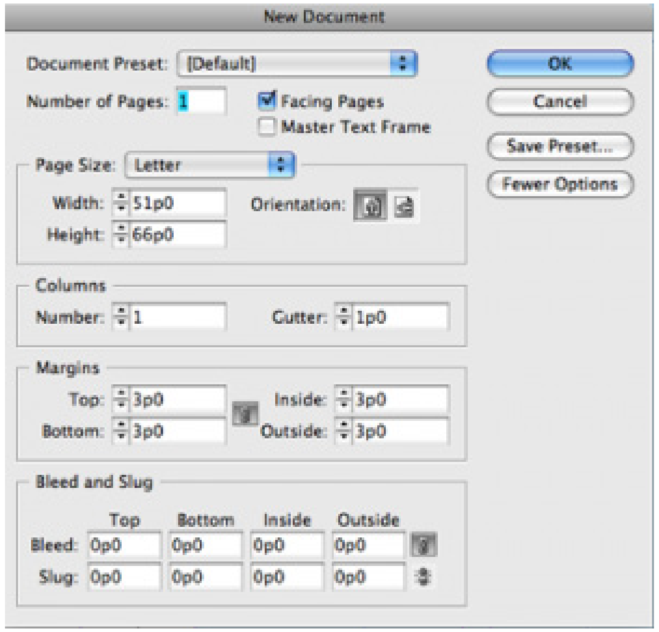
The nice thing about all these guide lines is that you can determine what color they appear in your document. And you can align those colors with the colors of your layers. Say your slug guide lines are light blue. You can create a layer called slug, make sure it is colored light blue, and then you can add whatever info to a text frame that you might find helpful. This makes the slug guidelines and your text frame the same color so they're easy to visually relate to each other.


Preferences — Guides & Pasteboard

Color

Margins: Magenta
Columns: Violet
Bleed: Fiesta
Slug: Grid Blue
Preview Background: Light Gray

And because InDesign is almost as flexible as a performer for Cirque de Soleil, you can either create all of these items when you create a new document:


New Document

Document Preset: [Default]

Number of Pages: 1 ✓ Facing Pages, Master Text Frame

Page Size: Letter — Width: 51p0, Height: 66p0, Orientation

Columns — Number: 1, Gutter: 1p0

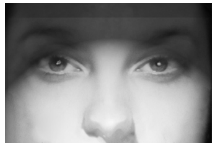
Margins — Top: 3p0, Bottom: 3p0, Inside: 3p0, Outside: 3p0

Bleed and Slug

	Top	Bottom	Inside	Outside
Bleed:	0p0	0p0	0p0	0p0
Slug:	0p0	0p0	0p0	0p0

You can have different margins on each spread, or even each page.

If you know you're going to have just a few different types of shell files with guides, swatches, layers, even styles then you can create a file that has all the info in it you want all of your documents to have, save it as a template (it will have .indt in the name). Then when you want to create a new file choose File/New/Document from Template. Open your desired file and you're ready to go and you'll know that all of your preferences are set. In fact, if you choose that option now you'll be taken to Bridge where you can see all the different templates that InDesign comes loaded with. Might be worth poking around to see if there is anything that will help you out.

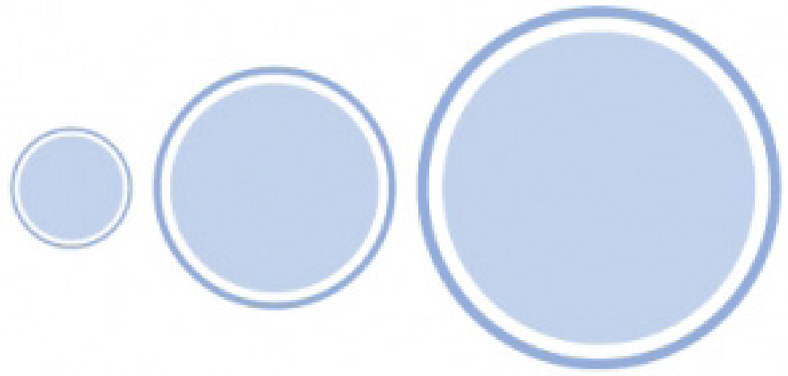


Stylin' Appearances

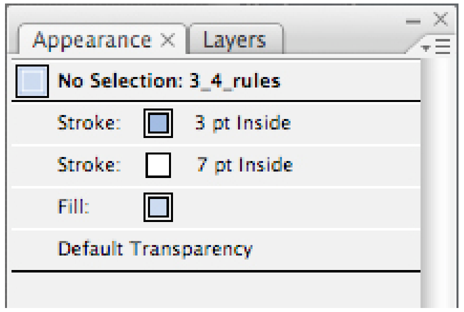
by Robin Storesund April 12, 2009

Graphic Styles are among my favorite tools in Illustrator because they give you so many options and can save you so much time. Styles in general are terrific — Graphic Styles, Character Styles, and Paragraph Styles — and I will go into detail on all of those some other time. Today I want to focus on graphic styles because I use those the most and I find them to be my biggest time savers. Plus, when you apply them to fonts they give you some really cool type effects and I just love me my type effects!

You can make graphic styles as simple or as complex as you need. Let's start with an apparently simple one — a circle with a 3 pt blue outer rule, a 4 pt white inner rule, and a tinted blue fill. You could actually draw concentric circles, but if you wanted to draw several different sized circles it would a time-consuming nuisance to have to calculate the sizes of the inner circles so that the proportion of the outer dark blue rule and the inner white space stayed the same. You can't just scale them because they will keep their relative proportions and resize thicker or thinner depending on which way you scale the circles, like so:



So the easiest way to maintain consistent proportions on the concentric circles is to create a style that you can then apply to circles of different sizes. The way you create such a style is by using the handy appearance panel, which I like to keep grouped with the layers panel since they present data in a similar manner. You start off with a single stroke and a single fill option but you can add additional strokes and fills to create the appearance you want, using the way the strokes and fills stack the same way you would use layers.



No Selection: 3_4_rules

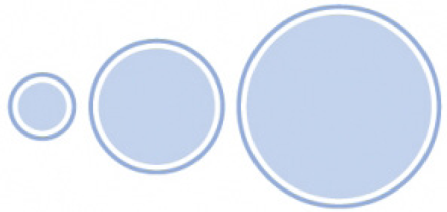
Stroke: 3 pt Inside

Stroke: 7 pt Inside

Fill:

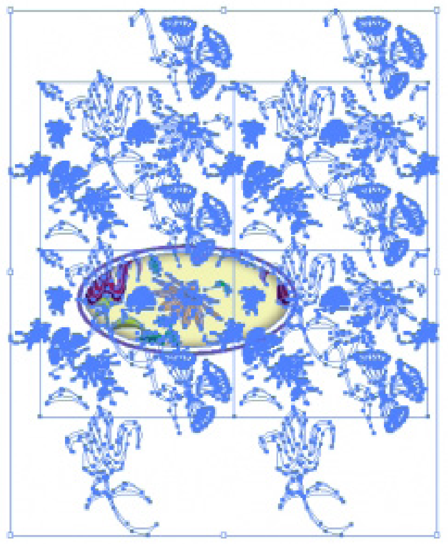
Default Transparency

To create a style for this appearance, I created a circle with a fill of 50% blue, a 7 pt white stroke aligned to inside, and a 3 pt. 100% blue stroke above that also aligned to inside. By placing the 3 pt. stroke on top of the 7 pt. stroke, that gives me a visible inner stroke of 4 pts. (7 – 3 = 4) and placing the fill underneath all of it creates the inner circle. Now the proportion of outer rule and inner rule stay the same no matter what size circle you apply it to, as you can see here:



Let's take this simple set of circles and create a style that's a little more fun. Instead of a solid color fill we're going to use a patterned fill and apply an effect to it. If you already have some patterned fills you can use those or you can load one of the many patterned fills included in Illustrator — I'm using "Wild Flowers Color" from the Nature_Foliage collection and I used a color from one of the flower groupings for the outer stroke. Now we're going to add an effect to make it more interesting. Select the fill, then go into the Effect menu and select Stylize, then select Inner Glow. Set Mode to Multiply and make the color 100% Black, set the Opacity to 60%, the Blur to 15 pts., and click on Edge.

As you can see, this gives the appearance of having a cutout over a print. The only thing you need to be aware of is if you are using a large tiled pattern fill, the area that is being used will change depending on where on the page it lies. If you have a particular area you want to use as your focal point you will need to expand the fill, which will also cause the rest of the pattern to appear under the clipping path.



Not only can you apply effects to the elements in a graphic style, you can also use the Transform effect to move one or more of the layers for even more fun, and these are the ones that can be so impressive when applied to type. In this week's download is a graphic style that is set up similar to the ones mentioned above, but I've also gone in and used the Transform effect to move the strokes down and to the right as well as adding a drop shadow to the fill. It's not so impressive when applied to a path, but with the right typeface, it looks like a famous sandwich cookie!

Once you have everything set up the way you want it, all you have to do to make it a graphic style is to select the path and drag it over into the Graphic Styles panel, then give it a descriptive name. To apply that new style to a path, just select the path and click on the name in the panel.

Stylin_Appearances.ai